Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #49359

BOARD # 81: WIP: Student outcomes as related to the interval between initial MATLAB instruction, potential interim programming encounters, and an intermediate MATLAB course

Dr. Jessica Thomas, The Ohio State University

Jessica Thomas is a Senior Lecturer at The Ohio State University, teaching first and second semester introductory engineering courses as well as an intermediate level MATLAB course.

WIP: Student outcomes as related to the interval between initial MATLAB instruction, potential interim programming encounters, and an intermediate MATLAB course

Introduction

The structure and timing of instructional material and courses has the potential for significant impacts on student outcome and retention of content. An example can be found in the practice of massed learning, where learning and study time is uninterrupted, versus spaced learning, where learners encounter the same material multiple times with spaces in between sessions [1]. The necessity for repetition in learning has been recognized for well over a century, as mentioned in the 1885 writings of Ebbinghaus, who observed that facts learned all at once before an exam and left unreinforced afterwards are soon forgotten [2], [3]. The body of literature further indicates that spaced learning produces better recall than massed learning [4], with the length of intervening events between encounters also influential in the retention of information over an extended period [4].

While acting as instructors for an intermediate level MATLAB course, we noticed a disparity between the performance of students, despite the assumption of a uniform introduction to the language. The majority of students in the first-year engineering program take Fundamentals of Engineering I, which introduces students to the language over a roughly 10 week period. However, students may also enroll in the course after taking an honors version of Fundamentals of Engineering which includes additional instruction in C/C++, or a transfer version of the course which can take place over a compressed timeline.

Further investigation revealed that the interval between students' introduction to MATLAB in their first-year courses and their continued education in the intermediate level course was also not uniform for all students. Without specific requirements for when to take the intermediate course, the interval between students' first collegiate introduction to the language and their intermediate course could potentially range from 1 month to more than 3 years. Moreover, depending on the student's major, they may have not utilized any programming languages in the interim, or may have written code on a regular basis in their courses. Some students may also have worked as teaching associates for the department, and so may have repeatedly used the language in an instructional capacity, even if it was not required for their major courses.

To better understand the disparity in performance and considering the influence of learning intervals on retention of material, we began collecting data on the various factors that may be impacting student grades to identify possible correlations. Specifically, we gathered data on the length of time since students' collegiate introduction to MATLAB, the circumstances of their introduction, their major, and potential teaching associate status versus their performance in their intermediate level course.

Ultimately, we wish to determine if the circumstances of introductory MATLAB courses and the interval between them and subsequent programming encounters impact student success in their second, intermediate MATLAB course. Ideally, these results would give students and program advisors guidance on the best timing of the intermediate course to maximize student success and retention of the course information.

Methods

Preliminary data was gathered via IRB approved surveys sent to students via email after the completion of their intermediate course. Students were allowed to take the surveys at a time and place of their choosing using their personal devices. Additional data is being gathered from institutional records, including grades, majors, time between introductory and intermediate MATLAB courses, and undergraduate teaching associate status. Data collection is still ongoing, but the preliminary response included information from 36 students.

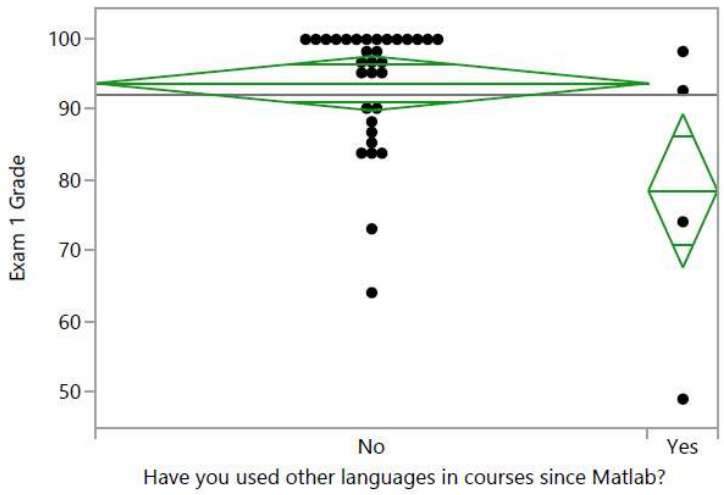
Students were asked for the following course information in the survey:

- Major
- Their introductory MATLAB course (this could be one of four courses, a standard course, an honors course, a scholars' course, and a transfer course)
- The semester of their introductory course
- Course location (main campus, regional campus, or local community college)
- Amount of prior programming experience
- If they had utilized MATLAB in their major courses since learning it
- Whether they had used other programming languages in their major courses since learning MATLAB
- If they had been an undergraduate teaching associate in our department (Engineering Education) and for how long
- Their approximate grades on the three intermediate course exams

Institutional records were then used to record students' actual grades on the three course exams and calculate their average exam score, and the data was deidentified. After de-identification, this information was analyzed using the JMP Student Edition statistical package and the scores compared to each of the factors listed above using a One-way ANOVA (note that this does not utilize the approximate exam grades).

Results

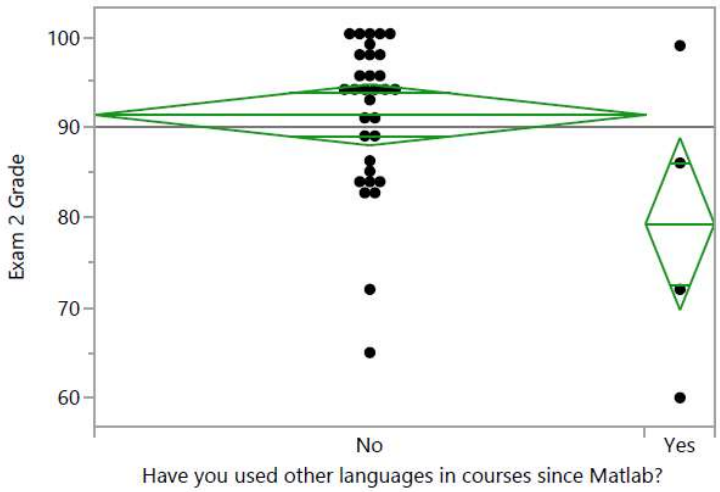
While the amount of data is limited and the results may change as more becomes available, the preliminary data showed an unexpected area of significance. Currently, there is a significant difference when examining students' individual and average exam scores versus whether they had used other programming languages in their major courses since learning MATLAB, as seen in Figures 1 through 4 below. Specifically, students who had used other languages in their major courses did not perform as well as students who did not use other languages.



Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Have you used other languages in courses since Matlab?	1	815.0703	815.070	7.1879	0.0112*
Error	34	3855.4297	113.395		
C. Total	35	4670.5000			

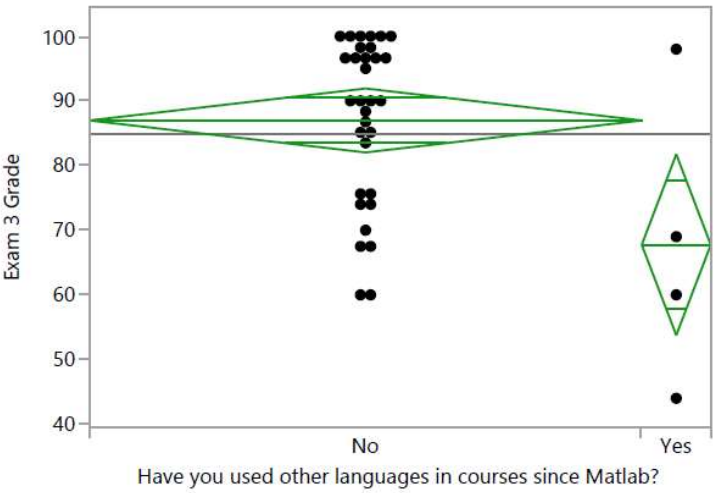
Figure 1: Oneway ANOVA comparing students' usage of other programming languages to their exam 1 score, demonstrating a significant difference (p=0.0112).



Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Have you used other languages in courses since Matlab?	1	518.6884	518.688	5.8923	0.0207*
Error	34	2992.9747	88.029		
C. Total	35	3511.6631			

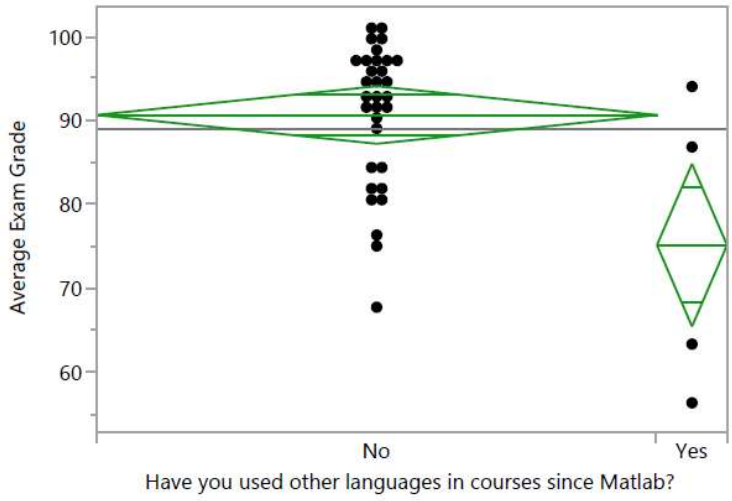
Figure 2: Oneway ANOVA comparing students' usage of other programming languages to their exam 2 score, demonstrating a significant difference (p=0.0207).



Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Have you used other languages in courses since Matlab?	1	1313.2813	1313.28	6.9289	0.0127*
Error	34	6444.2188	189.54		
C. Total	35	7757.5000			

Figure 3: Oneway ANOVA comparing students' usage of other programming languages to their exam 3 score, demonstrating a significant difference (p=0.0127).



Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Have you used other languages in courses since Matlab?	1	855.2559	855.256	9.4009	0.0042*
Error	34	3093.1872	90.976		
C. Total	35	3948.4431			

Figure 4: Oneway ANOVA comparing students' usage of other programming languages to their average exam score, demonstrating a significant difference (p=0.0042).

Another trend that was not significant, but was consistent across the exams, was the clustering of scores in certain majors, with Aerospace Engineering consistently performing well overall relative to other majors, as seen below in figure 5.

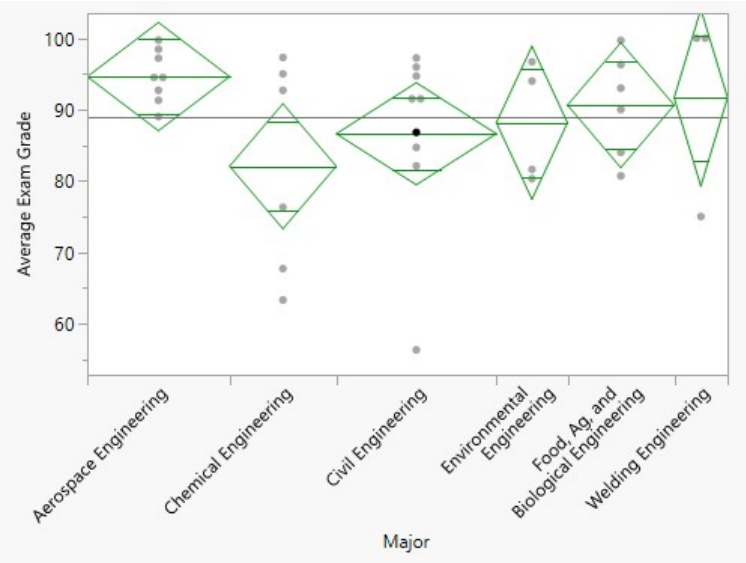


Figure 5: Oneway ANOVA comparing students' major versus their average exam score. Though this pattern occurred across all exams, the average is shown above for reference.

Discussion and Future Directions

Upon first examination, the results in figures 1-4 seem paradoxical, as it might be assumed that experience in more than one programming language would lead to stronger programming performance overall. However, with the context that the students in this category had no programming experience prior to their introductory MATLAB course, these results make more sense. Since programming languages are processed much like spoken languages [5], an interrupted, incomplete understanding of one language while attempting to learn a second could easily lead to confusion of the syntax between the two. If this result is borne out by a larger data set, it may imply that care should be taken to schedule programming courses so that a higher level of mastery is obtained in one language prior to starting the second, or that specific care and instruction is needed when studying two languages simultaneously.

Though not statistically significant, the consistent clustering and performance of students majoring in Aerospace Engineering is also notable and presents an opportunity to examine the differences in frequency and timing of programming exposure between majors. While it is possible that this phenomenon is due to sampling, it also raises the possibility of curriculum differences between majors playing a role in course performance.

Although difficult to draw conclusions with the current limited data set, it will be interesting to see if a significant pattern continues to exist between simultaneous language learners and emerges between majors. No significant effect of the timing of the two courses or other examined factors has been observed yet, though it remains to be seen if that will change as the data set grows. If a difference emerges between majors, it may in fact be indirectly due to timing, given the differences in course scheduling between different curricula.

Ultimately, it will be important to see if these patterns continue, and if any additional differences emerge so that students and departments can be advised on the best strategy to support student success.

.

References

[1] N. J. Cepeda, H. Pashler, E. Vul, J. T. Wixted, and D. Rohrer, “Distributed practice in verbal recall tasks: A review and quantitative synthesis.,” *Psychol. Bull.*, vol. 132, no. 3, pp. 354–380, 2006, doi: 10.1037/0033-2909.132.3.354.

[2] H. Ebbinghaus, *Memory: a contribution to experimental psychology*. New York: Dover Publications, 1964.

[3] “Memory: A Contribution to Experimental Psychology,” *Ann. Neurosci.*, vol. 20, no. 4, pp. 155–156, Oct. 2013, doi: 10.5214/ans.0972.7531.200408.

[4] A. w. Melton, “The situation with respect to the spacing of repetitions and memory,” *J. Verbal Learn. Verbal Behav.*, vol. 9, no. 5, pp. 596–606, Jan. 1970, doi: 10.1016/S0022-5371(70)80107-4.

[5] N. Peitek *et al.*, “A Look into Programmers’ Heads,” *IEEE Trans. Softw. Eng.*, vol. 46, no. 4, pp. 442–462, Apr. 2020, doi: 10.1109/TSE.2018.2863303.